In the claims:

Following is a complete set of claims as amended with this Response.

1. (Currently Amended) A method, comprising:

receiving a persistence package from one of a plurality of different software components, the persistence package including persistent data and metadata, the software components having persistent data in different formats;

extracting persistent data and metadata from the persistence package, the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component, the metadata describing the persistent data including a class file with bytecodes for each object;

establishing, based on the extracted metadata, a transform for a storage format for the persistent data during a runtime of a ~~of the~~ receiving system;

applying the transform to the persistent data to format the persistent data without using the software component from which the persistence package was received during the runtime of the receiving system from the format of the software component into a storage format that is compatible with the receiving system and with a storage device independent of the software component; and

storing the persistent data in the storage device in the storage format during the runtime of the system.

2. (Previously Presented) The method of claim 1, further comprising using metadata passed from the persistence package to establish a storage location for the persistent data during the runtime of the system.

3. (Previously Presented) The method of claim 1, wherein the metadata comprises at least in part a description of a model structure of the persistent data.

4. (Previously Presented) The method of claim 3, wherein the metadata conforms to a metadata template comprising rules for describing the model structure.

5. (Previously Presented) The method of claim 4, wherein extracting the persistent data and the metadata from the persistence package comprises using a filter.

6. (Canceled)

7. (Original) The method of claim 1, further comprising retrieving persistent data from storage using a transform during the runtime of the receiving system.

8. (Original) The method of claim 1, further comprising receiving persistent data compatible with at least one of any type of processor, any type of programming language, any type of operating system, and any type of architecture.

9. (Currently Amended) An apparatus, comprising:

a data storage device;

a receiving system coupled to the data storage device, the receiving system including a persistence engine to receive a persistence package from one of a plurality of different software components, the persistence package including persistent data and metadata, the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component, the metadata describing the persistent data including a class file with bytecodes for each object, the software components having persistent data in different formats, wherein the persistence engine extracts persistent data and metadata from the persistence package, wherein the persistence engine uses the extracted metadata passed from the persistence package to establish, without using the software component from which the persistence package was received, during a runtime of the receiving system, a storage format to store the persistent data in the data storage device, and wherein the persistence engine applies the storage format to the persistent data to format the persistent data during the runtime of the receiving system from the format of the software component into a storage format that is compatible with the receiving system and with the storage device independent of the software component.

10. (Previously Presented) The apparatus of claim 9, wherein the data storage device is external to a receiving system using the persistence engine.

11. (Previously Presented) The apparatus of claim 9, further comprising a storing interface to store the persistent data using the storage format.

12. (Previously Presented) The apparatus of claim 9, further comprising a retrieving interface to retrieve stored persistent data for use by one of the receiving system and the software component, the software component comprising an application.

13. (Original) The apparatus of claim 9, wherein the metadata comprises at least in part a description of the data model structure of the persistent data.

14. (Original) The apparatus of claim 13, further comprising a metadata template to format the metadata for readable reception by the persistence engine.

15. (Original) The apparatus of claim 9, wherein the persistence engine receives a persistence package comprising the metadata and the persistent data.

16. (Original) The apparatus of claim 9, wherein the persistence engine receives persistent data structured using any data model from a source comprising at least one of any type of processor, any type of operating system, any type of programming language, and any type of architecture.

17. (Original) The apparatus of claim 9, further comprising a metadata engine having a metadata reader and a metadata filter.

18. (Original) The apparatus of claim 17, wherein the metadata filter interprets the metadata.

19. (Original) The apparatus of claim 9, further comprising a transform engine having a set of transforms, a transform selector, and a transform generator.

20. (Original) The apparatus of claim 19, wherein a transform establishes at least one of the storage format and the storage location to store the persistent data in the data storage device.

21. (Original) The apparatus of claim 19, the transform selector further comprising a data model comparator.

22. (Original) The apparatus of claim 19, wherein the transform selector selects a transform based on filtered metadata.

23. (Original) The apparatus of claim 19, wherein the transform selector requests a transform from the transform generator based on filtered metadata.

24. (Original) The apparatus of claim 23, wherein the transform generator produces a transform that remodels the persistent data to approximate as closely as possible a preexisting transform from the set of transforms.

25. (Previously Presented) The apparatus of claim 23, wherein the transform generator produces a transform that substantially maintains the model structure of the persistent data received by the receiving system.

26. (Original) The apparatus of claim 23, wherein the transform generator produces a transform to remodel the persistent data to maximize efficient retrieval for an application.

27. (Original) The apparatus of claim 23, wherein the transform generator uses iterative read-write trials to produce a transform to remodel the persistent data to maximize storage and/or retrieval speed.

28. (Original) The apparatus of claim 23, wherein the transform generator produces a transform to remodel the persistent data to maximize data compression.

29. (Currently Amended) An apparatus, comprising:

a communications interface;

a data model description receiver to receive a data model description from one of a plurality of different software components, the software components having persistent data in accordance with different data models;

a set of transforms;

a data model comparator to produce a comparison independent of the software component from which the data model description is received between the data model description and a data model in a transform in the set of transforms;

a transform generator, operational during runtime of the apparatus ~~a system~~, having an assembler to produce a transform based on the data model description and the comparison independent of the software component from which the data model description was received;

a storage device; and

a transform engine to apply a transform to format persistent data for storage from the format of the software component into a storage format that is compatible with the ~~with a~~ storage device independent of the software component.

30. (Previously Presented) The apparatus of claim 29, wherein the transform generator further comprises a data model variance calculator coupled to the assembler.

31. (Previously Presented) The apparatus of claim 29, wherein the transform generator further comprises a data model approximator coupled to the assembler.

32. (Previously Presented) The apparatus of claim 29, wherein the transform generator further comprises an efficient storage/retrieval speed maximizer coupled to the assembler.

33. (Previously Presented) The apparatus of claim 32, wherein the storage/retrieval speed maximizer further comprises a read/write iterator.

34. (Previously Presented) The apparatus of claim 29, wherein the transform generator further comprises a data compression maximizer coupled to the assembler.

35. (Previously Presented) The apparatus of claim 29, wherein the transform generator further comprises an indexing estimator coupled to the assembler.

36. (Currently Amended) A method, comprising:

receiving a data model description from one of a plurality of different software components, the software components having persistent data in accordance with different data models, the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component, the persistent data being associated with metadata describing the persistent data including a class file with bytecodes for each object;;

comparing the data model description to a preexisting data model independent of the software component from which the data model description is received;

assembling a transform independent of the software component from which the data model description is received based on the data model description and the

comparison to establish a storage format for persistent data during runtime of a system; and

applying a transform to format persistent data for storage from the format of the software component into a storage format that is compatible with a storage device independent of the software component.

37. (Original) The method of claim 36, wherein the assembling a transform includes measuring a variance between the data model description and a preexisting data model.

38. (Original) The method of claim 36, wherein the assembling a transform includes approximating a preexisting data model.

39. (Original) The method of claim 36, wherein the assembling a transform includes maximizing data storage speed and/or data retrieval speed.

40. (Original) The method of claim 39, wherein the maximizing speed includes iteratively performing data read/write trials and selecting the fastest trial.

41. (Original) The method of claim 36, wherein the assembling a transform includes maximizing data compression.

42. (Original) The method of claim 36, wherein the assembling a transform includes optimizing efficient indexing for the persistent data.

43. (Currently Amended) An article of manufacture, comprising:

a machine-readable medium comprising instructions, that when executed cause a machine to:

receive persistent data having a model structure from one of a plurality of different software components, the software components having persistent data in different model structures, the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component;

receive metadata comprising at least in part a description of the model structure, the metadata describing the persistent data including a class file with bytecodes for each object; and

establish, using the metadata and without using the software component from which the persistence package was received,-during a runtime of the machine, a storage format for the persistent data; and

apply the established storage format to the persistent data to format the persistent data for storage from the format of the software component into a storage format that is compatible with the machine and with a storage device independent of the software component.

44. (Previously Presented) The article of manufacture of claim 43, further comprising instructions, that when executed, cause a machine to store the persistent data using the storage format.

45. (Original) The article of manufacture of claim 43, further comprising instructions, that when executed, cause a machine to receive metadata conforming to a metadata template comprising rules for describing a data model structure of the persistent data.

46. (Previously Presented) The article of manufacture of claim 45, further comprising instructions, that when executed, cause a machine to receive a persistence package comprising the persistent data and the metadata and to extract the persistent data and the metadata from the persistence package.

47. (Previously Presented) The article of manufacture of claim 43, further comprising instructions, that when executed, cause a machine to retrieve the persistent data using the storage format.

48. (Original) The article of manufacture of claim 43, further comprising instructions, that when executed, cause a machine to select and/or create, based on the metadata, a transform to establish at least one of the storage format and the storage location.

49. (Original) The article of manufacture of claim 43, further comprising instructions, that when executed, cause a machine to receive persistent data compatible with one of any type of processor, any type of programming language, any type of operating system, and any type of architecture.